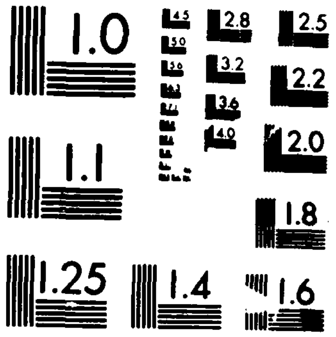
AD-A183 330 PERFORMANCE EVALUATION OF DECISIONMAKING ORGANIZATIONS 1/1
(U) MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR
INFORMATION AND DECISION SYSTEMS H P HILLION ET AL.
UNCLASSIFIED JUL 87 LIDS-P-1683 N00014-84-K-0519 F/G 12/6 NL


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

JULY 1987 LIDS-P-1683

PERFORMANCE EVALUATION OF DECISIONMAKING ORGANIZATIONS*

by

Herve P. Hillion**
Alexander H. Levis***

ABSTRACT

Decisionmaking organizations are modeled as asynchronous concurrent systems using Timed Petri Nets. This model allows the evaluation of time-related performance with respect to the following measures: the maximum throughput rate, defined as the maximum processing rate achievable by the system, and the execution schedule, which determines the earliest instants at which the different operations can occur in the process. These measures of performance (MOPs) are expressed as functions of the time and resource constraints that are in force in any particular organization. The characterization obtained makes it possible to compare different organizational forms and to modify existing designs so as to improve performance. The methodology is illustrated through the analysis of a five-member ship control party of a submarine.

*This work was carried out at the MIT Laboratory for Information and Decision Systems with support by the Office of Naval Research under Contract No. N00014-K-0519 and by the Joint Directors of Laboratories through the Office of Naval Research under Contract No. N00014-85-K-0782.

**INRIA-Lorraine, Vandoevre, France.

***Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

PERFORMANCE EVALUATION OF DECISIONMAKING ORGANIZATIONS*

by

Herve P. Hillion — INRIA-Lorraine, Vandoevre, FRANCE

Alexander H. Levis — Laboratory for Information and Decision Systems, MIT, Cambridge, MA

ABSTRACT

Decisionmaking organizations are modeled as asynchronous concurrent systems using Timed Petri Nets. This model allows the evaluation of time-related performance with respect to the following measures: the maximum throughput rate, defined as the maximum processing rate achievable by the system, and the execution schedule, which determines the earliest instants at which the different operations can occur in the process. These measures of performance (MOPs) are expressed as functions of the time and resource constraints that are in force in any particular organization. The characterization obtained makes it possible to compare different organizational forms and to modify existing designs so as to improve performance. The methodology is illustrated through the analysis of a five-member ship control party of a submarine.

I. INTRODUCTION

The performance of a decisionmaking organization is affected by the internal structure of the organization and by the time and resource constraints present. The internal structure, through the communication and execution protocols, determines which activities must be sequential and which can be concurrent. The organization members, individually, may have resource constraints due to cognitive limitations such as limited memory, or to hardware limitations such as channel capacity, or to software limitations such as lack of multi-tasking capability. Furthermore, the processing times associated with the various activities are finite; they can be modeled as time constraints that affect the ability of the organization to produce results in a timely manner.

Timed Petri Nets can be used to analyze the dynamic behavior of systems with asynchronous and concurrent processing. So far, two models of Timed Petri Nets have been studied: Deterministic models [1,2,3,4] in which the transition firing times are assumed to be fixed, and probabilistic models [5,6,7,8], where the firing times are specified by probabilistic distribution functions, generally assumed to be exponential distributions. In both cases, methods are developed to evaluate the steady state performance. A more general approach is proposed in this paper. The transition firing times are considered to be dependent on the number of firing repetitions. The method can handle a sequence of successive firing times for every transition; the sequence may be either deterministic or random. The analysis is restricted, however, to a special class of Petri Nets, namely Event-Graphs [9], and is focused on obtaining performance measures. These results are well suited to the performance evaluation of tactical organizations supported by command, control and communications (C^3) systems.

In Section 2, the assumptions of the model are presented, while in Section 3, an upper bound to the average firing rate is computed, that depends only on the average firing time of the transitions. In Section 4, a fast and simple algorithm that determines the firing schedule, when the firing times are deterministic, is presented. Finally, the results are used in Section 5 to analyze a five person decisionmaking organization, the ship control party of a submarine.

II. TIMED EVENT GRAPHS

An Event Graph [9,11] (also known as Marked Graph [12]) is a Petri Net in which each place has exactly one input and one output transition (Fig. 1). Given an initial distribution of tokens in the net (i.e., an initial marking), an Event-Graph is live if and only if the number of tokens in every directed elementary circuit is strictly positive [12]. We assume here that this condition is satisfied so that each transition can fire repeatedly any number of times.

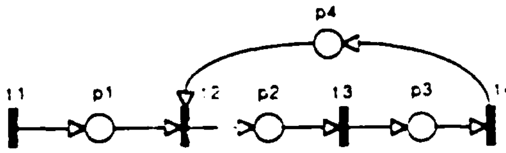


Figure 1. An Event Graph

*This work was carried out at the MIT Laboratory for Information and Decision Systems with support by the Office of Naval Research under Contract No. N00014-K-0519 and by the Joint Directors of Laboratories through the Office of Naval Research under Contract No. N00014-85-K-0782.

In Timed Petri Nets, each transition t takes a "real" time $\mu(t)$ to fire. When a transition t is enabled, a firing can be initiated by removing one token from each of t's input places. The tokens remain in transition t for the firing execution during the time $\mu(t)$ and then the firing terminates by adding one token in each of t's output places. Note that in the case of a general underlying Petri Net, arcs can be integer valued, although we assume here that all arc values are unity.

Different models of Timed Petri Nets have been studied; in deterministic Timed Petri Nets [1,2,4] and Timed Event-Graphs [3], a positive (rational) number is assigned to each transition t of the net, which defines the firing time $\mu(t)$. In stochastic Timed Petri Nets [5,6,7], the firing times are assumed to be random variables that are exponentially distributed.

In this paper, the transition firing times are not fixed, but may be different from one firing to the next. Therefore, a sequence of successive firing times $\{\mu_t(1), \mu_t(2), \ldots, \mu_t(k), \ldots\}$ (for any transition t) is constructed according to the number of firing repetitions. There is no assumption regarding this sequence, except that the following limit:

$$\lim_{n \to \infty} \frac{\sum_{k=1}^{n} \mu_t(k)}{n} = \bar{\mu}_t \qquad (1)$$

must exist and be finite and non-null for any transition t. The limit, denoted by $\bar{\mu}_t$, determines the average (or mean) firing time of transition t. Note that, in practice, the firing time of a transition corresponds to the processing of a task. Therefore, the existence of a mean processing time is a rather realistic assumption for most physical systems.

The sequence of successive firing times can be regarded as possible outcomes of the random variable μ_t, with mean value $\bar{\mu}_t$ (regardless of the probability distribution function). For instance, μ_t can be a discrete random variable that takes on a finite set of possible values $\{v_1, v_2, \ldots, v_J\}$ according to a certain probability distribution $\{\gamma_1, \ldots, \gamma_J\}$. In that case:

$$\bar{\mu}_t = \sum_{k=1}^{J} \gamma_k \, v_k$$

This assumption is useful in the performance evaluation of decisionmaking organizations. In this case, the processing of a task can be performed by different algorithms, each having a fixed execution time. At each occurence of the task, an algorithm is selected, according to a fixed probability distribution: such a decision rule is modeled by a switch [13] in the Petri Net model.

In order to study the performance of Event-Graphs, it is assumed in this paper that the Event-Graph model is strongly connected and live. These assumptions ensure that the transition firings can be repeated any number of times and that the net is bounded [14]. The following notation will be used:

$T = \{t_1, t_2, \ldots, t_m\}$ is the set of transitions.

$\mu_i(n)$ denotes the n-th firing time of transition t_i (i.e., when t_i fires for the n-th time).

$S_i(n)$ denotes the instant at which the n-th firing initiation of transition t_i occurs.

M_i^0 denotes the initial marking (number of tokens at the initial time) in place i.

It is assumed, without loss of generality, that the initial instant is $\tau = 0$ and that $S_i(0) = 0$ and $\mu_i(0) = 0$ for $i = 1, 2, \ldots, m$. The dynamic behavior of the system starting from $\tau = 0$ will be described by the sequence $S_i(n)$ for $i = 1, 2, \ldots, m$ and $n = 1, 2, \ldots$, which will also be called the firing schedule. Therefore, it is assumed that transitions fire as soon as they are enabled: in the case of Timed Event Graphs, we know that this rule leads to an optimal repetitive schedule, in the sense that we obtain the earliest firing schedule (see [3] for proof). Hence the performance obtained will be the maximum performance with respect to time. Note that for general nets (other than Event Graphs) such an earliest schedule does not exist and the above rule is not necessarily optimal.

III. PERFORMANCE EVALUATION

In this section, an upper bound to the performance in steady-state is obtained. The performance measure considered is the average period π_i with which any transition t_i fires, i.e.,

$$\pi_i = \lim_{n \to \infty} \frac{\sum_{k=1}^{n} (S_i(k) - S_i(k-1))}{n} \qquad (2)$$

Quite obviously, $1/\pi_i$ determines also the average firing rate of transition t_i. We note that π_i can also be written as:

$$\pi_i = \lim_{n \to \infty} \frac{S_i(n)}{n} \qquad (3)$$

since $S_i(0) = 0$ by assumption. Note that the limit of $S_i(n)$ as n goes to infinity is unbounded. This follows from the assumption that the average firing times are strictly positive.

Since the Event-Graph is assumed to be strongly connected, the average period is the same for all transitions of the net and will be denoted by π. This is trivially deduced by the fact that the number of tokens in any directed elementary circuits is invariant with any transition firing [12]. A directed elementary circuit is a directed path that goes from one node (place or transition) back to itself and such that none of the nodes are repeated. Figure 1 illustrates such a directed elementary circuit with three transitions and three places that we denote by $\rho = \{t_2 p_2 t_3 p_3 t_4 p_4\}$.

The average cycle time of any directed elementary circuit ρ, denoted by $C(\rho)$, is defined as the sum of the average firing times of all transitions belonging to the circuit divided by the number of tokens in the circuit

$$C(\rho) = \frac{\sum_{i \in I} \bar{\mu}_i}{N} \tag{4}$$

where $\bar{\mu}_i$ is the average firing time of transition t_i $(i=1,\ldots,r)$ as defined by relation (1) and N is the total number of tokens in the circuit as determined by:

$$N = \sum_{j=1}^{r} M_j^{\circ} \tag{5}$$

i.e., as the sum of the initial marking of all places p_j in the circuit ρ $(j=1,2,\ldots,r)$. Recall that in an Event Graph, N is invariant by any transition firings and, therefore, is time invariant.

Now let C_{max} be the maximum over all directed elementary circuits of the average cycle times. C_{max} will be called the maximum average cycle time.

$$C_{max} = \max_{\rho} C(\rho) \tag{6}$$

Then the following result holds (for proof, see Hillion [10]):

Theorem The maximum average cycle time is a lower bound of the average period, i.e.,

$$\pi \geq C_{max} \tag{7}$$

This theorem generalizes the result first obtained by Ramchandani [1] and extended further in [3] in which all transition firing times were deterministic and are constant.

The relationship (7) also yields an upper bound to the average firing rate:

$$1/\pi \leq 1/C_{max} \tag{8}$$

This bound can be computed quite easily, once all circuits are determined. A simple method for obtaining all circuits is described in [10], using an algorithm that determines the invariants of a net [15]. It is particularly interesting that the upper bound of the maximum performance only depends of the transition mean firing times (given the initial distribution of tokens), regardless of any other assumptions concerning the firing times (which may be either deterministic or random variables with any type of probability distribution). Note that (8) is equality (i.e., $\pi = C_{max}$) when the transition firing times are constant [1], and that the steady-state is then K-periodic [3].

In computing C_{max}, it is also interesting to determine the critical circuits, i.e., those circuits ρ for which $C(\rho) = C_{max}$. It turns out that only these circuits bound the average firing rate in steady-state. Accordingly, the critical circuits should be the ones to modify (in terms of transition firing times or number of tokens in the circuit) so as to improve the performance of a system.

We are now going to determine the firing schedule, assuming that the sequences of successive firing times are known, i.e., the firing process is deterministic.

IV. DETERMINATION OF THE FIRING SCHEDULE

We present in this section a fast algorithm for computing the firing schedule, i.e., the sequence $S_i(n)$ $i=1,\ldots,m$; $m=1,2,\ldots$, when the system is deterministic. An initial distribution of tokens is assumed to be given at $\tau = 0$ and the sequence of successive firing times $\mu_i(n)$ $n=1,2,\ldots$ is assumed to be fixed for any transition t_i. We first determine the order with which transitions fire (regardless of the actual time they fire) which we call partial firing order.

Given the structure of the net and the initial distribution of tokens, some transitions fire sequentially and others fire concurrently. In order to specify how the firings occur in the process, we proceed as follows. We call marked places the places which contain one token. We first consider the set P_1 of places that contain at least one token, i.e., with an initial marking strictly positive, and take this set as the initial set of marked places. Then let T_1 be the set of transitions that are enabled by the places of P_1. Assume that all transitions of T_1 fire once and let P_2 be the new set of marked places. Let T_2 be the set of transitions that are enabled by the places of P_2 and which do not have already fired (i.e., do not belong to T_1). Let P_3 be the set of marked places once all transitions of T_2 have fired. In a similar way, we then consider the set T_3 of transitions that are enabled by P_3 and which do not have fired already (i.e., do not belong to $T_1 \cup T_2$) and so on. We repeat this operation s times until $T_{s+1} = \phi$. Because the net is a live Event-Graph, the sequence $T_1, T_2, \ldots, T_s$ so constructed verifies:

$$T = T_1 \cup T_2 \ldots \cup T_s$$

where T denotes the set of all transitions of the net. This sequence determines therefore the partial firing order between the transition firings. In particular, the transitions belonging to T_i $(i=1,\ldots,s)$ are transitions that fire concurrently at the i-th step. For that reason, the sequence $T_1, T_2, \ldots, T_s$ will be called the sequence of concurrent transitions sets. It should be clear that this sequence is fully determined by the structure of the net, and the initial distribution of tokens.

For clarity, we now assume that the transitions are labeled according to the sequence $T_1, \ldots, T_s$, i.e.,

$$T_1 = \{t_1, t_2, \ldots, t_{k1}\}$$

$$T_2 = \{t_{k1+1}, t_{k1+2}, \ldots, t_{k2}\}$$

$$\ldots$$

$$T_s = \{t_{ks+1}, \ldots, t_m\}$$

where m denotes the total number of transitions in the net. Note that the order between the transitions belonging to T_i $(i=1,\ldots,s)$ does not matter, since these transitions fire concurrently.

For any transition t_j we now denote by $Inp(t_j)$ the set of all transitions which are the input transitions of all input places of t_j, as shown on Figure 2; $\{p_{i1}, p_{i2}, \ldots, p_{ir}\}$ are all input places of transition t_j and $\{t_{i1}, t_{i2}, \ldots, t_{ir}\}$ is the set of input transitions of each of these places. If $t_i \in Inp(t_j)$, we also denote by M^0_{ij} the initial marking of the unique place p_{ij} whose input transition is t_i and output transition t_j.

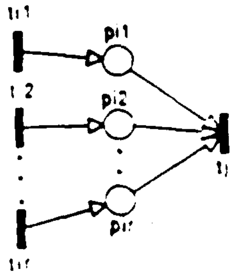


Figure 2. Input Places and Transitions of t_j

For any transition t_j and any transition $t_i \in Inp(t_j)$ we have:

$$S_j(n+M^\circ_{ij}) \geq S_i(n) + \mu_i(n) \qquad (9)$$

where n is any positive integer. Suppose that $n > M^\circ_{ij}$. Then (9) can be written as:

$$S_j(n) \geq S_i(n-M^\circ_{ij}) + \mu_i(n-M^\circ_{ij}) \qquad (10)$$

For simplicity, let us assume $S_i(k) = 0$ and $\mu_i(k) = 0$ if $k \leq 0$ (for $i=1,2,\ldots,m$), so that (10) is still satisfied if $n \leq M^\circ_{ij}$; this means that there are still tokens left in the place p_{ij} from the initial marking, so that the firing initiations of t_j do not depend on the firing terminations of t_i.

Given the assumption that the firings occur as soon as the transitions are enabled, we deduce from (10) the following relation:

$$S_j(n) = \max_{t_i \in Inp(t_j)} (S_i(n-M^\circ_{ij}) + \mu_i(n-M^\circ_{ij})) \qquad (11)$$

Equation (11) can now be used to compute the firing schedule by iterating on n, the number of firing repetitions. At the n-th iteration, we can compute $S_j(n)$ $(j=1,2,\ldots,m)$ once we know both $S_i(k)$ for $i=1,2,\ldots,m$ and $k=1,2,\ldots,n-1$ and $S_i(n)$ for $i=1,2,\ldots,j-1$. Consider indeed any transition $t_i \in Inp(t_j)$. If $M^\circ_{ij} > 0$, we then know $S_i(n-M^\circ_{ij})$ (computed at a previous interation). Now if $M^\circ_{ij}=0$, it means that in the partial firing order, t_i should necessarily fire before t_j. Therefore, if $t_j \in T_v$ then $t_i \in T_u$ with $u < v$. Since we have assumed that the transitions are labeled according to the partial order defined by the sequence $T_1, T_2, \ldots, T_s$, it follows that $i < j$. Finally, if $t_j \in T_1$ then, by construction, M°_{ij} is always strictly positive whenever $t_i \in Inp(t_j)$.

The algorithm can now be described recursively in a very simple way:

```
Initialization

p = Max ($M^\circ_{ij}$)
for k = -p to 0, set $S_i(k) = 0$ and $\mu_i(k) = 0$
(for i=1,...,m)
n = 1 (n is the number of firing repetitions)
Repeat (main loop)
For j = 1 to m   set

    $S_j(n) = \max_{t_i \in Inp(t_j)} (S_i(n-M^\circ_{ij}) + \mu_i(n-M^\circ_{ij}))$

End when n=R (total number of firing repetitions).
```

Recall that in order to use the algorithm, there are two steps to complete:

(1) Determine the firing order of the transitions, following the method described in this section.

(2) Determine the set $Inp(t_j)$ for any transition t_j (deduced from the structure of the net).

This polynomial algorithm of complexity O(m R) has been used to compute the firing schedule of decisionmaking organizations, using a Timed Event-Graph model [10]. It should be noted that, in this approach, we do not need to construct a state graph.

V. APPLICATION

In this section, an example of a five DM organization is used to illustrate the results. The structure of the organization is fixed, but the resource and time constraints vary. In each case considered, the execution schedule and the maximum throughput rate are computed. The measures of performance obtained are then analyzed and compared. The task processing times considered in the examples are all deterministic.

The Petri Net representation of the organizational structure considered is shown in Fig. 3. The five DM organization models a generalized submarine ship control party performing emergency control tasks. This particular organization has been extensively described and analyzed in [16], using the information theoretic framework as a way to evaluate the workload of the organization members and the accuracy of the response. The goal here is to evaluate the time-related performance of the organization.

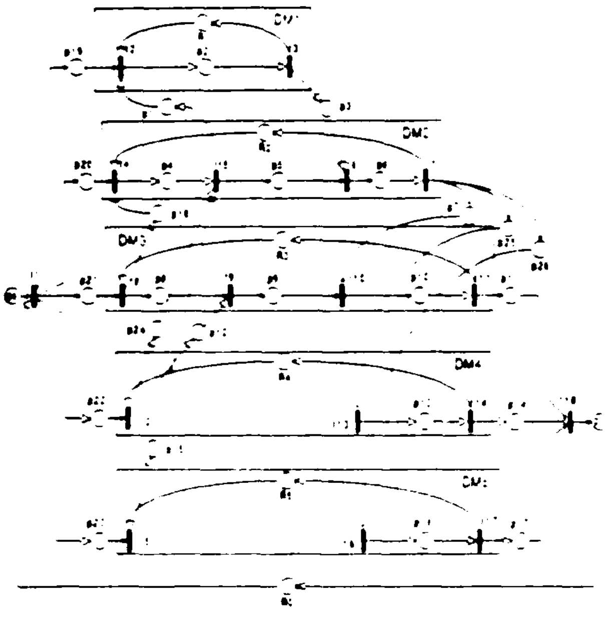


Figure 3. Model of the Five DM Organization with Limited Resources

The organization has both hierarchical and parallel aspects, where the DMs play in fact different roles. DM1 acts as a supervisor, DM2 acts as a coordinator and DM3, DM4 and DM5 act as subordinates. Indeed, DM3, DM4 and DM5 transmit their situation assessments to DM2 and receive commands inputs from this DM. However, before issuing these command, DM2 fuses the information transmitted by these three DMs with his own assessment and transmits the resulting information to DM1, the supervisor. DM1 combines this information with his own situation assessment and then produces a command to DM2. DM2 proceeds to select a response that he then transmits back to the subordinates as a command. Finally, the latter select in turn a response and the three produce the output which constitutes the organization's response.

Resource places have been added between the RS stage and the SA stage of each decisionmaker between the output transition and the input transition of the net.

Both the directed circuits and the slices of the Petri Net of Fig. 3 are determined from the Incidence Matrix. The columns of the incidence matrix, C, correspond to the transitions of the net and the rows to the places. The element C_{ij} of the matrix is such that:

$$C_{ij} = \begin{cases} -1 & \text{if } p \text{ is an } \underline{\text{input}} \text{ place of } t_j \\ +1 & \text{if } p_i \text{ is an } \underline{\text{output}} \text{ place of } t_j \\ 0 & \text{otherwise} \end{cases}$$

The Incidence Matrix corresponding to the Petri Net model of Figure 3 has 18 transitions, 26 places and 6 resource places. The directed circuits were obtained using an implementation of the algorithm developed by Alaiwan and Toudic [17]. The total number of circuits is fifty-two (52) and every circuit contains at least one of the resource places. Therefore, the organizational form is admissible, i.e., the information structure is acyclical. There are eleven slices, shown in Table 1, that model the different steps of the complete process.

The maximum throughput rate and the execution schedule of the organization will now be determined for different values of the resource and time constraints.

TABLE 1. LIST OF THE SLICES

Slice 1:	t1
Slice 2:	t8 , t12, t15
Slice 3:	t9 , t4
Slice 4:	t5
Slice 5:	t2
Slice 6:	t3
Slice 7:	t6
Slice 8:	t7
Slice 9:	t10, t13, t16
Slice 10:	t11, t14, t17
Slice 11:	t18

Measures of Performance

The analysis is based on the following information:

The transition firing times, corresponding to the task processing times, and the initial marking of the resource places, corresponding to the resources available for processing.

The firing schedule which corresponds to the sequence (S_i^n), for i = 1,2,...,m (total number of transitions in the net.

The critical circuits.

The following measures of performance (MOPs) can be computed from this information:

(a) The maximum throughput rate, Φ, given by:

$$\Phi = K/\pi \tag{12}$$

(b) The dynamic response time corresponding to the complete processing of N inputs:

$$T_m^N = S_m^N + \mu_m \tag{13}$$

where t_m is the output transition of the net. The ratio

$$RT_N = T_m^N/N \tag{14}$$

gives an approximate value of the average response time RT (defined as the limit of RT_N when N goes to infinity). Recall that:

$$RT = 1/\Phi \tag{15}$$

(c) The average processing time (of any individual input), defined as:

$$d = \frac{d_n + d_{n+1} + \ldots + d_{n+K-1}}{K} \tag{16}$$

where:

$$d_n = T_m^n - S_1^n = S_m^n - S_1^n + \mu_m \tag{17}$$

and:

$$d = M^0(R_0)/\Phi \tag{18}$$

where $M^0(R_0)$ denotes the initial marking of the resource place of the overall organization.

Results

Four different cases of resource and time constraints were analyzed. In the first case, the organization (DMO) as a whole and each of the five DMs can only handle one input at a time. All the task processing times are assumed identical and equal to one unit of time. In the second case, there are more resources available for processing: the DMO can handle four external inputs at the same time and each DM is able to process at most two inputs simultaneously. In the third case, the resources available are assumed to be the same as in the second example, but the task processing times of DM4 are equal to two units of time. Finally, the fourth case is an example where the processing times are arbitrary; the resources available are identical and set to three (the DM and the organization can handle three inputs at a time).

In each case, the firing schedule has been computed for a total of ten (N=10) repetitions of the process. Since the total number of transitions in the net is 18, the firing schedule obtained corresponds to sequence (S_i^n) for i = 1,2,...,18 and n = 1, 2,...,10. The algorithm described in section 4 has been used to determine this sequence.

The results obtained for the four cases are summarized on Table 2.

TABLE 2. SUMMARY OF THE RESULTS

MOP	CASE 1	CASE 2	CASE 3	CASE 4
π	11.0	9.0	11.0	25.5
K	1	2	2	3
Φ	0.091	0.222	0.182	0.118
T_m^{10}	110.0	48.0	59.0	102.0
RT_{10}	11.0	4.8	5.9	10.2
RT	11.0	4.5	5.5	8.5
d	11.0	18.0	22.0	25.5

For the first case, the maximum throughput rate is 0.091 inputs processed per unit of time. The three critical circuits obtained have eleven transitions each and the token content is one. Furthermore, they all have one unique resource place, which is R_0. By deleting R_0, we obtain the directed paths for which the processing time is maximal: d is 11 units of time and the dynamic response time, corresponding to the processing of the ten inputs, is 110 units of time.

The results for the second case show that the steady-state process is 2-periodic with a period π of 9 units of time. Accordingly, the maximum throughput rate is 0.222 and, in that case, the dynamic response time is 48 units of time. The ratio RT_{10}, which is equal to 4.8 units of time per input, is quite close to the average response time RT, which is 4.5 units of time per input. The average response time is 18. Finally, there is a unique critical circuit, which includes nine transitions and its token content is two, corresponding to the initial marking of the resource place R_2.

In the third case, the steady-state process is 2-periodic with a period of π of 11 units of time and the maximum throughput rate is, therefore, 0.182. The process reaches its steady-state after the fifth repetition (N=5) and the dynamic response time is 59.0 units of time and the average processing time is 22.

As in the second case, there is a unique critical circuit, which is internal to the organizational structure: the interactions between DM1, DM2 and DM4 are the ones that constrain the maximum throughput rate. The resources of DM4 are now critical.

In the fourth case, the transition firing times are totally arbitrary and the initial marking of all the resource places is set to three tokens. The steady-state process is 3-periodic with a period π of 25.5 units of time. Accordingly, the maximum throughput rate is 0.118 inputs processed per units of time, the process reaches its steady-state after the fourth repetition, the dynamic response time is 102.0 units of time, and the average procesing time is 25.5 units of time.

The performance measures, summarized in Table 2, indicate that the first case is the worst one. The maximum throughput rate is the lowest and the dynamic response time T_m^N (resp. the average response time RT) is the longest. In the second case, where there are more resources available for processing, the performance is greatly improved. The maximum throughput rate is more than twice the one obtained in the first case, while the response time is less. However, the average processing time, d, which represents the average amount of time it takes to process any individual input, is now larger. In the third case, the resources available for processing are the same as in the second case, but the processing times of DM4 are twice as long as in the other ones. As expected, the performance obtained is slightly degraded, when compared to the second case, but is still much better than in the first case. Recall that in the fourth case, the task processing times are arbitrary (but are greater or equal to one unit of time) and there are more resources than in the first three cases. It turns out that the performance obtained is worse than in case 2 and 3, but is still better than in case 1. The critical circuits, obtained in the four cases determine precisely which of the time and resource constraints bound the throughput rate.

At this point, the execution schedule obtained for the concurrent tasks, i.e., the instants of time at which the transitions within a slice fire, is very interesting to analyze. In cases 1 and 2, all the transitions within a slice fire at the same instants of time (at each repetition of the process), which is not true for case 3 and 4. This is not surprising when all the transition firing times are equal to one unit of time. However, when the transition firing times are arbitrary, there is no reason why the concurrent operations should occur at the same instant of time. This is actually what makes the process to occur asynchronously in real-time.

From the execution schedule it is possible to identify which operations take place (repeatedly) at the latest instants, compared to the other concurrent operations. For instance, in case 3, the transitions that fire the latest are t_{11} in $\bar{T}_3$ and t_{14} in $\bar{T}_{10}$. These transitions are critical for the processing delays. It follows that, by reducing their processing times, the time-delay of the organization will improve. By determining from the execution schedule the concurrent tasks that fire at the latest instants, we identify which time constraints are critical for the processing delays and, therefore, it becomes easier to improve the related performance of the organization.

VI. CONCLUSION

We have developed in this paper some techniques for analyzing real-time systems that can be modeled using Event-Graphs. The result presented in Section 3 generalizes the result obtained in [1] for constant transition firing times. It can be used for preliminary performance evaluation of a general C^3 system for which only the average task execution times are known. It holds, in particular, without the restrictive assumptions of the probabilistic models studied so far. We have also extended the deterministic case, allowing a different firing time at each repetition. The algorithm described in section 4 provides a simple way to obtain the precise firing schedule for this case without the need for simulation. These results extended the set of tools available for the analysis and design of command and control organizations.

VII. REFERENCES

[1] Ramchandani, C. (1974). "Analysis of asynchronous concurrent systems by Timed Petri Nets", Technical Report No. 120, Laboratory for Computer Science, MIT, Cambridge, MA.

[2] Sifakis, J. (1980). "Performance Evaluation of Systems using Nets", Net Theory and Applications, Lecture Notes in Computer Science, Springer-Verlag, Berlin, FRG.

[3] Chretienne, P. and Carlier, J. (1984). "Modeling scheduling problems with Timed Petri Nets," Advances in Petri Nets, 1984, Lecture Notes in Computer Science, No. 188, Springer-Verlag, Berlin, FRG, pp. 62-82.

[4] Ramamoorthy, C.V. and G.S. Ho (1980). "Peformance Evaluation of asynchronous concurrent systems using Petri Nets," IEEE Trans. Software Eng., Vol. SE-6, No. 5, pp. 440-4[illegible].

[5] Zuberek, V. M. (1985). "M-Timed Petri Nets, priorities, preemptions, and performance evaluation of systems," Advances in Petri Nets, 1985, Lecture Notes in Computer Science, No. 222, Springer-Verlag, Berlin, FRG, pp. 478-498.

[6] Molloy, M.K., (1982). "Performance analysis using stochastic Petri Nets," IEEE Trans. on Computers, Vol. 31, No. 9, pp. 913-917.

[7] Natkin, S. (1985). "Timed and Stochastic Petri Nets: From the validation to the performance of synchronization schemes," Proc. Workshop on Timed Petri Nets, Torino, Italy, pp.2-3.

[8] Wiley, R.P. (1986). "Performance analysis of stochastic Timed Petri Nets," Ph.D. Thesis, LIDS-TH-1525, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

[9] Brams, G.V. (1983). Reseaux de Petri: theorie et pratique, Masson, Paris, France.

[10] Hillion, H. P. (1986). "Performance evaluation of Decisionmaking Organizations using Timed Petri Nets," M.S. Thesis, LIDS-TH-1590, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

[11] Peterson, J. L. (1981). Petri Net theory and the modeling of systems, Prentice-Hall, Englewood Cliffs, NJ.

[12] Commoner, F., A. W. Holt, S. Even, and A. Pnueli, (1971). "Marked Directed Graphs," Journal of Computer and System Sciences, Vol. 5, No. 5, pp. 511-523.

[13] Tabak, D. and A. H. Levis (1985). "Petri Net representation of decision models," IEEE Trans. on Systems, Man and Cybernetics, Vol. 5, SMC-15, No. 6, pp. 812-818.

[14] Cohen, G., D. Dubois, J.P. Quadrat and M. Viot (1985). "A linear-system theoretic view of discrete event processes and its use for performance evaluation in manufacturing," IEEE Trans. on Automatic Control, Vol. Ac-30, No. 3, pp. 210-220.

[15] Martinez, J. and M. Silva (1980). "A simple and fast algorithm to obtain all invariants of a generalized Petri Net," Lecture Notes in Computer Science, No. 52, Springer-Verlag, Berlin, pp. 302-310.

[16] Weingaertner, S.T., (1986). "A Model of Submarine Emergency Decisionmaking and Decision Aiding," S.M. Thesis, LIDS-TH-1612, Laboratory for Information and Decision Systemns, MIT, Cambridge, MA.

[17] Alaiwan,H., and J.M. Toudic,(1985). "Recherche des Semi-Flots, des Verrous et des Trappes dan les Reseaux de Petri," Technique et Science Informatiques, Vol. 4, No. 1, Dunod, France, pp. 103-112.

END
9-87
DTIC